

## Project 1 - ETL with Airflow

This project consists of an ETL pipeline built with Apache Airflow containerised in Docker and utilising PostgreSQL.

### Project Overview

This project demonstrates an ETL pipeline using Apache Airflow containerised in Docker.

The project consists of the following:

- Extracts data from
- Transforms data into a load-able format (data cleaning if applicable)
- Loads transformed data into a PostgreSQL Database through Airflow
- Uses DAGs inside Airflow to schedule and perform tasks

### Tools Used:

Workflow: Apache Airflow

Containerisation: Docker

Data Source: REST Countries API (<https://restcountries.com>)

Data Store: PostgreSQL

Documentation: README.md and .pdf

Project Management: Trello

### Pipeline Design:

graph

### Setting up the environment:

#### 1. Clone the Repository from GitHub

```
git clone https://github.com/guidavids/Project-1—Airflow.git
```

```
cd "Project 1 - Airflow"
```

#### 2. Run in your preferred environment

Open with PyCharm (or any other IDE)

Set up a virtual environment with PyCharm or your preferred Python interpreter to read and run the .py files

### 3. Start Docker with the .yaml file

Type “docker compose up” on any terminal when Docker Desktop is running. When done with the “-d” flag it will run in disjointed format

Note: Type “docker compose down” for when you want to stop and delete the containers whilst retaining the images

### 4. Access Airflow UI

Open your preferred browser and enter “http://localhost:8080” when Docker is running or click 8080:8080 on the “containers” section of Docker Desktop

Airflow Login details:

- username: airflow - password: airflow

## Project Management

This project was managed using **Trello**

The tasks were as follows: - Data Source: Find Dataset - Data Store: Choose Database - Set up Docker Desktop - Set up Docker images and containers - Create the main Diagram - Create the Database Table - Create the ETL Pipeline - Extract data from API - Transform and clean data (if applicable) - Load the data into a .csv file - Load data into Postgres with Airflow dags